STAT 4360 (Introduction to Statistical Learning, Fall 2022)
Mini Project 4

Name: Haeun Kim

1

(a)

I got the RMSE 1.065438 and the test MSE is 1.135158 by using the caret package, in addition, I got same test MSE when I used the cv.glm

(b)

By using the best-subset selection based on adjusted R^2, I could figure out the RMSE is 0.9330443, and the test MSE is 0.8705717, in addition, I got same test MSE when I used the cv.glm.

(c)

By using the forward stepwise selection based on adjusted R^2, I could figure out the RMSE is 0.9330443, and the test MSE is 0.8705717, in addition, I got same test MSE when I used the cv.glm.

(d)

By using the backward stepwise selection based on adjusted R^2, I could figure out the RMSE is 0.9330443, and the test MSE is 0.8705717, in addition, I got same test MSE when I used the cv.glm.

(e)

By using the ridge regression with penalty parameter chosen optimally via LOOCV, the best lambda is 0.3764936, and the test MSE by using this lambda is 0.7110372.

(f)

By using the lasso regression with penalty parameter chosen optimally via LOOCV, the best lambda is 0.1232847, and the test MSE by using this lambda is 0.7107995.

(g)

I would recommend Lasso regression because the test MSE for lasso regression is the least one among others. In addition, lasso regression says that the coefficient of clarity and body is zero.

	Linear regression	Best-subset	Forward stepwise	Backward stepwise	Ridge regression	Lasso regression
Test MSE	1.135158	0.8705717	0.8705717	0.8705717	0.7110372	0.7107995
Intercept	7.81437	8.1208167	8.1208167	8.1208167	7.5937659	7.842919240
Clarity	0.01705				0.1177222	0
Aroma	0.08901				0.2420135	0.006591392

Body	0.07967				0.2073347	0
Flavor	1.11723	1.1920393	1.1920393	1.1920393	0.8120571	1.064840570
Oakiness	-0.34644	-0.318316	-0.318316	-0.318316	-0.295769	-0.12734765
Region2	-1.51285	-1.515484	-1.515484	-1.515484	-1.304990	-1.31495263
Region3	0.97259	1.0935478	1.0935478	1.0935478	0.9038720	1.069376007

2.

(a)

I got the accuracy 0.7785193 and the test error rate is 0.2214807 by using the caret package

(b)

By using the best-subset selection based on AIC, I could figure out accuracy of this model is 0. 7800303, and the test error rae is 0.2199697

(c)

By using the forward stepwise selection based on AIC, I could figure out accuracy of this model is 0. 7800303, and the test error rae is 0.2199697

(d)

By using the backward stepwise selection based on AIC, I could figure out accuracy of this model is 0. 7800303, and the test error rae is 0.2199697

(e)

By using the ridge regression with penalty parameter chosen optimally via 10-folds cross validation, the best lambda is 0.01, and the confusion matrix is below

	0	1
0	1180	136
1	306	378

So, the accuracy is 0.779, so the test error rate is 0.221.

(f)

By using the lasso regression with penalty parameter chosen optimally via 10-folds cross validation, the best lambda is 0.01, and the confusion matrix is below

	0	1
0	1180	136
1	315	369

So, the accuracy is 0.7745, so the test error rate is 0.2255.

(g)

I could recommend one of best-subset, forward stepwise, and backward stepwise. This is because the test error rates of them are the least one. Furthermore, they dropped skinthickness, which was dropped in project 3 because it didn't have significant relationship by p-value. This result is consistent with the previous project, so I would recomment one of best-subset, forward stepwise, and backward stepwise.

	Linear regression	Best-subset	Forward stepwise	Backward stepwise	Ridge regression	Lasso regression
Test error	0.2214807	0.2199697	0.2199697	0.2199697	0.221	0.2255
Intercept	-8.026451	-8.027315	-8.027315	-8.027315	-7.497432	-7.234208
Pregnancy	0.1263845	0.126371	0.126371	0.126371	0.114258	0.1063796
Glucose	0.0337202	0.033681	0.033681	0.033681	0.0306523	0.0301612
Bloodpressure	-0.009645	-0.009581	-0.009581	-0.009581	-0.008047	-0.003939
Skinthickness	0.0005185				0.000346	0
Insulin	-0.001243	-0.001212	-0.001212	-0.001212	-0.000890	-0.0003713
BMI	0.0775549	0.077874	0.077874	0.077874	0.0704422	0.06227519
Pedigree	0.8877583	0.889495	0.889495	0.889495	0.8085093	0.63083315
Age	0.0129414	0.012894	0.012894	0.012894	0.0142498	0.01023284

```
library("caret")
library("ISLR2")
library("leaps")
library("glmnet")
library("bestglm")
library("boot")

setwd("C:/Users/haeun/OneDrive/문서/STAT33550")

#Bringing the wine dataset
wine <- read.table("wine.txt", header = TRUE)
totpred <- ncol(wine) - 1
#Factoring the Region from wine dataset
wine$Region = as.factor(wine$Region)

#Question 1-(a)
#The linear regression for the quality by using the LOOCV
#Using the caret library to calculate the LOOCV error rate
full <- lm(Quality~Clarity + Aroma + Body + Flavor + Oakiness + Region, data = wine)
summary(full)
ctrl <- trainControl(
  method = 'LOOCV',
  number = 1
)
loocv_caret <- train(Quality~Clarity + Aroma + Body + Flavor + Oakiness + Region, data = wine,
method = "lm", trControl = ctrl)
loocv_caret$results
#RMSE 1.065438
1.065438^2
# Calculate the test MSE by using the cv.glm function

glm.fit = glm(Quality ~., data=wine)
cv.error = cv.glm(wine, glm.fit, K = nrow(wine))$delta[1]
mean(cv.error)

#Question 1-(b)
#Using the best-subset selection based on adjusted R^2
fit.full <- regsubsets(Quality ~ ., wine, nvmax = totpred)
fit.summary <- summary(fit.full)
fit.summary$adjr2

# the maximum adjusted R2 is 0.8164362
max_adjr2 <- which.max(fit.summary$adjr2)
coef(fit.full, max_adjr2)

# To calculate the test MSE by using the caret package
loocv_subselec <- train(Quality~Flavor + Oakiness +
```

```
                  Region, data = wine,
                 method = "lm", trControl = ctrl)
loocv_subselec$result
# RMSE 0.9330443
0.9330443^2
# test MSE = 0.8705717

#Figuring out the test MSE again by using the cv.glm function with coefficient
glm.fit = glm(Quality~Flavor + Oakiness +
          Region, data=wine)
cv.error = cv.glm(wine, glm.fit, K = nrow(wine))$delta[1]
mean(cv.error)

#Question 1-(c)
#Using the forward step-wise selection based on adjusted R^2
fit.fwd <- regsubsets(Quality ~ .,
                data = wine, nvmax = totpred,
                method = "forward"
)
fit.summary.fwd <- summary(fit.fwd)

#To figuring out the best adjusted R^2
fit.summary.fwd$adjr2
max_adjr2_fwd <- which.max(fit.summary.fwd$adjr2)
coef(fit.fwd, max_adjr2_fwd)

#Training with predictors we figured out above
loocv_subselec <- train(Quality~Flavor + Oakiness +
                  Region, data = wine,
                 method = "lm", trControl = ctrl)
#And calculate the test MSE
loocv_subselec$result
# RMSE 0.9330443
0.9330443^2
# test MSE = 0.8705717

#Figuring out the test MSE again by using the cv.glm function with coefficient
glm.fit = glm(Quality~Flavor + Oakiness +
          Region, data=wine)
cv.error = cv.glm(wine, glm.fit, K = nrow(wine))$delta[1]
mean(cv.error)

#Question 1-(d)
# By using the backward step-wise selection based on adjusted R^2
# to find the best linear regression model
fit.bwd <- regsubsets(Quality ~ .,
                data = wine, nvmax = totpred,
                method = "backward"
```

```
)

fit.summary.bwd <- summary(fit.bwd)

#To find out the best adjusted R^2
fit.summary.bwd$adjr2
max_adjr2_bwd <- which.max(fit.summary.bwd$adjr2)
#Figuring out the coefficient for the best linear regression model
coef(fit.bwd, max_adjr2_bwd)

# fit the best linear regression model to find out the test MSE
loocv_subselec <- train(Quality~Flavor + Oakiness +
                          Region, data = wine,
                        method = "lm", trControl = ctrl)
loocv_subselec$result
# RMSE 0.9330443
0.9330443^2
# test MSE = 0.8705717

#Question 1-(e)

# Y is the Quality from the wine data set
y <- wine$Quality
# X has all the variables except for the Quality, which is response
x <- model.matrix(Quality ~ ., wine)[, -1]

# Making the grid and use it as lamba for glmnet function(ridge regression model)
grid = 10^seq(10, -2, length = 100)
ridge_mod <- glmnet(x, y, alpha = 0, lambda = grid)
# Setting the seed
set.seed(1)
# by using cross-validation with nfold(due to LOOCV) figuring out the best lambda
cv.out <- cv.glmnet(x, y, alpha = 0, nfolds = nrow(wine), lambda = grid)
bestlam <- cv.out$lambda.min
#Best Lambda is 0.3765
ridge.pred <- predict(ridge_mod, newx = x, s = bestlam)
mean((ridge.pred-y)^2)
#Figuring out the coefficients for this regression model
predict(ridge_mod, type = "coefficients", s = bestlam)[1:8, ]


#Question 1-(f)
# Y is the Quality from the wine data set
y <- wine$Quality
# X has all the variables except for the Quality, which is response
x <- model.matrix(Quality ~ ., wine)[, -1]

# Making the grid and use it as lamba for glmnet function(lasso regression model)
```

```
grid = 10^seq(10, -2, length = 100)
lasso_mod <- glmnet(x, y, alpha = 1, lambda = grid)
# Setting seed with 1
set.seed(1)

# by using cross-validation with nfold(due to LOOCV) figuring out the best lambda
cv.out <- cv.glmnet(x, y, alpha = 1, nfolds = n, grouped = FALSE,lambda = grid)
bestlam <- cv.out$lambda.min
#Best Lambda is 0.1232847
lasso.pred <- predict(lasso_mod, newx = x, s = bestlam)
mean((lasso.pred-y)^2)
#Figuring out the coefficients for this regression model
predict(lasso_mod, type = "coefficients", s = bestlam)[1:8, ]

#####################################################################
#Bringing the diabetes dataset
diabetes <- read.csv("diabetes.csv", header = TRUE)
totpred <- ncol(diabetes) - 1

#Question 2-(a)
diabetes$Outcome <- as.factor(diabetes$Outcome)
#Logistic regression model for all the predictors from diabetes dataset
fit <- glm(Outcome ~ Pregnancies.. + Glucose.. + BloodPressure.. + SkinThickness.. +
          Insulin..  + BMI.. + DiabetesPedigreeFunction.. + Age.., family = binomial, data = diabetes)
summary(fit)
# Using caret package, cross-validation with 10 folds to find out the error rate
ctrl <- trainControl(
  method = 'CV',
  number = 10
)
set.seed(1)
cv10_caret <- train(Outcome~Pregnancies.. + Glucose.. + BloodPressure.. + SkinThickness.. +
                    Insulin..  + BMI.. + DiabetesPedigreeFunction.. + Age.., data = diabetes, method =
"glm", trControl = ctrl)
#The accuracy for this model
cv10_caret$result
1-0.7785193
#test error rate is 0.2214807

#Question 2-(b)
#Using the best-subset selection based on AIC
diabetes$y = diabetes$Outcome
diabetes$Outcome = NULL
bglm.AIC = bestglm(Xy = diabetes, family = binomial, IC = "AIC")
bglm.AIC$BestModel
# the AIC is 1930
set.seed(1)
diabetes <- read.csv("diabetes.csv", header = TRUE)
```

```
diabetes$Outcome <- as.factor(diabetes$Outcome)
cv_subselec <- train(Outcome ~ Pregnancies.. + Glucose.. + BloodPressure.. +
                       Insulin.. + BMI.. + DiabetesPedigreeFunction.. + Age.., data = diabetes,
                     method = "glm", trControl = ctrl)
cv_subselec$result
# Accuracy is 0.7800303
1-0.7800303
# test error rate = 0.2199697

#Question 2-(c)
diabetes <- read.csv("diabetes.csv", header = TRUE)
totpred <- ncol(diabetes) - 1
diabetes$Outcome <- as.factor(diabetes$Outcome)
diabetes$y = diabetes$Outcome
diabetes$Outcome = NULL
#To find best model by using the bestglm package
bglm.AIC = bestglm(Xy = diabetes, family = binomial, IC = "AIC", method = "forward")
bglm.AIC$BestModel
# the AIC is 1930
set.seed(1)
diabetes <- read.csv("diabetes.csv", header = TRUE)
diabetes$Outcome <- as.factor(diabetes$Outcome)
cv_subselec <- train(Outcome~Pregnancies.. + Glucose.. + BloodPressure.. +
                       Insulin.. + BMI.. + DiabetesPedigreeFunction.. + Age.., data = diabetes,
                     method = "glm", trControl = ctrl)
cv_subselec$result
# The Accuracy is 0.7800303
1 - 0.7800303
# The test error rate is 0.2199697

#Question 2-(d)
diabetes <- read.csv("diabetes.csv", header = TRUE)
diabetes$Outcome <- as.factor(diabetes$Outcome)
diabetes$y = diabetes$Outcome
diabetes$Outcome = NULL
#To find best model by using the bestglm package based on AIC
bglm.AIC = bestglm(Xy = diabetes, family = binomial, IC = "AIC", method = "backward")
bglm.AIC$BestModel
# the AIC is 1930
set.seed(1)
diabetes <- read.csv("diabetes.csv", header = TRUE)
diabetes$Outcome <- as.factor(diabetes$Outcome)
cv_subselec <- train(Outcome~Pregnancies.. + Glucose.. + BloodPressure.. +
                       Insulin.. + BMI.. + DiabetesPedigreeFunction.. + Age.., data = diabetes,
                     method = "glm", trControl = ctrl)
cv_subselec$result
# The Accuracy is 0.7800303
1 - 0.7800303
```

```
# The test error rate is 0.2199697

#Question 2-(e)
# n is the number of row of diabetes data set
n <- nrow(diabetes)
# Y is the Outcome from the diabetes data set
y <- diabetes$Outcome
# X has all the variables except for the Outcome, which is response
x <- model.matrix(Outcome ~ ., diabetes)[, -1]

# Making the grid and use it as lamba for glmnet function(ridge regression model)
grid = 10^seq(10, -2, length = 100)
ridge_mod <- glmnet(x, y, alpha = 0, lambda = grid, family = "binomial")

#Setting seed with 100
set.seed(100)
# by using cross-validation with 10-folds figuring out the best lambda
cv.out <- cv.glmnet(x, y, alpha = 0, nfolds = 10, grouped = FALSE,lambda = grid, family = binomial)
bestlam <- cv.out$lambda.min
#Best Lambda is 0.01
#Making the confusion matrix to figure out the accuracy
confusion_matrix <- table(y, predict(ridge_mod,newx = x, type = "class", s = bestlam))
confusion_matrix
#Predicted coefficient
predict(ridge_mod, type = "coefficients", s = bestlam)[1:9, ]


#Question 2-(f)
# Y is the Outcome from the diabetes data set
y <- diabetes$Outcome
# X has all the variables except for the Quality, which is response
x <- model.matrix(Outcome ~ ., diabetes)[, -1]

# Making the grid and use it as lamba for glmnet function(lasso regression model)
grid = 10^seq(10, -2, length = 100)
lasso_mod <- glmnet(x, y, alpha = 1, lambda = grid, family = "binomial")

# Setting seed with 100
set.seed(100)
# by using cross-validation with 10-folds figuring out the best lambda
cv.out <- cv.glmnet(x, y, alpha = 1, nfolds = 10, grouped = FALSE,lambda = grid, family = "binomial")
bestlam <- cv.out$lambda.min
#Best Lambda is 0.01
#Making the confusion matrix to figure out the accuracy
confusion_matrix <- table(y, predict(lasso_mod,newx = x, type = "class", s = bestlam))
confusion_matrix
#Predicted coefficient
predict(lasso_mod, type = "coefficients", s = bestlam)[1:9, ]
```